

Segmentation Process

15 & 16 January 1998

Mr. Robert P. Blaisdell
Electronic Systems Center DII AF
email: rpb@mitre.org
(781) 271-5757

Background

DII COE Engineering Segmentation Guidance

- **Must Continue Support to Self-Describing Segment Concept**
- **Must be Designed for Windows NT and 95 Logo Compliant**
 - Must Support Smooth Transition to Native NT v5.0 and ZAW Capabilities
 - Work in conjunction with the "add software" Control Panel Applet
 - Must Use the NT Registry as Information Store for Designed for Windows NT and 95 Logo and DII COE Required Data Entries
- **Must be Capable of Flexible Installation COTS, GOTS and Shareware**
- **Must Support Stand-Alone, Enterprise and Web Based Installation Mechanisms**

NT

Advisory

DII COE Engineering Guidance Segmentation

- **Concept of "Self-Describing" Segments is a DII COE Requirement that Should be Continued on the NT Platform**
- **Augmentation of the DII COE Guidance Documents is needed to Address Logo Program Specifications**
 - **Includes the I&RTS, the Style Guide, and Others**
- **In Evaluating and Recommending Augmentation of the COE with the Logo Program, it is understood that:**
 - **Extensions Unique to Microsoft are to be Avoided When Possible**
 - **When not possible, need must be Clearly Articulated with Supporting Rationale for DISA Engineering to make such Decisions**

DII COE Engineering Guidance

Installation/De-Installation

- **"Segment" Install/De-install is a Common Function and the DII COE Should Continue to be Responsible for this Function as Part of the NT-COE Kernel**
- **"Common" Install/de-Install Capability must be:**
 - **Compliant with the Design for Windows NT and 95 Logo Program Specifications**
 - **Should Transition Smoothly to the Native Install/De-Install Capabilities Planned for NTv5 and the Zero Administration Workstation.**

DII COE Engineering Guidance

Installation-De-Installation

- **Some features that the Install/De-Install must support include:**
 - **Stand-alone, enterprise (push from a server), and "web-based" installations**
 - **Attended and un-attended installations**
 - **Work "in conjunction with" the NT "add software" Control Panel Applet**
 - **Capability to install/de-install COTS, GOTS, and most distribution formats of Shareware (ZIP, Self-exploding EXE, etc.)**

Logo Handbook Section

6.2

Installation

Requirements/Recommendations

~~User Experience~~

- Using the Registry
- System & Shared Components
- Install Support of a Networked or Managed Environment
- Use of Policy in a Managed Environment
- Supporting Migration from Windows 95 to Windows NT
- Win95 to Win NT 5.0 Application Migration DLL

Logo Handbook Section

6.3

Uninstall

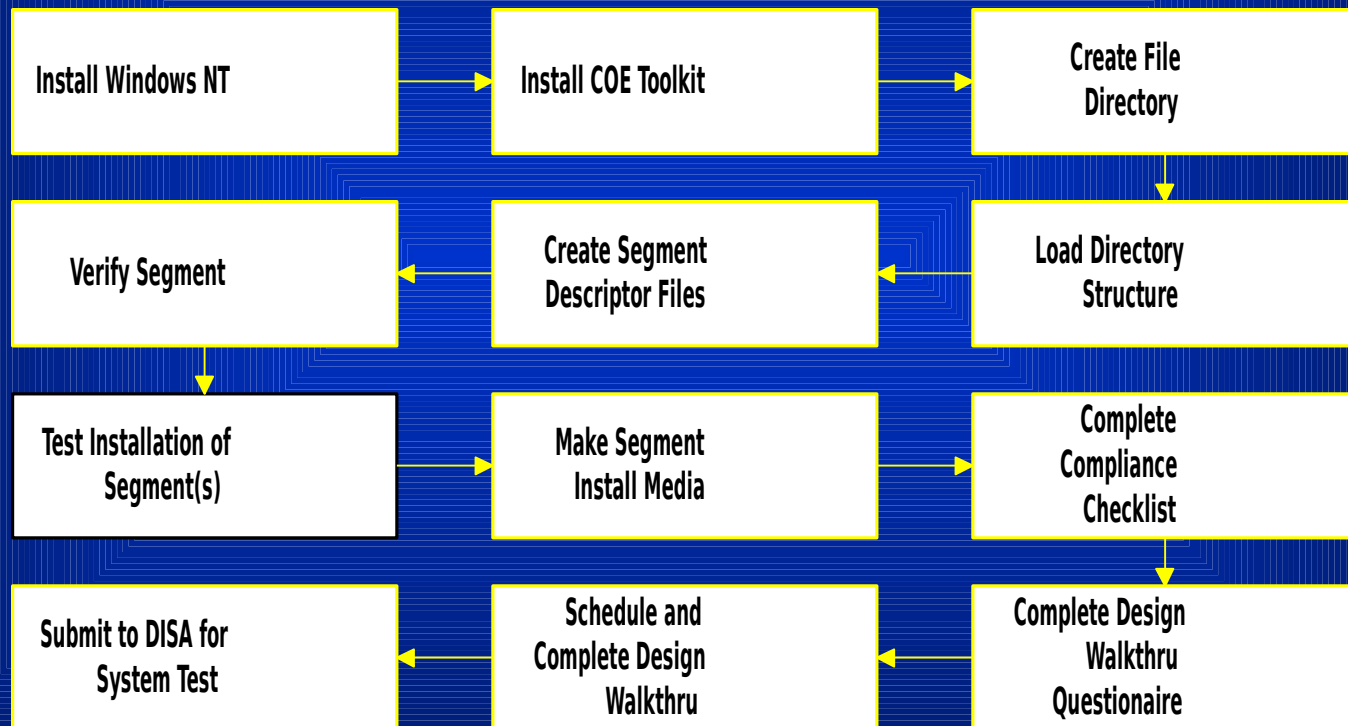
Requirements/Recommendations

- User Experience
- Registry, System & Shared Components
- Uninstall Support of a Networked or Managed environment
- Managed Installation Service - coming with NT 5.0

Current Segmentation Process (v3.0 -3.3)

- **Create Segment Descriptor Files (SDFs) for Applications, as well as following the Logo Program Requirements for Installation and Deinstallation.**
- **Determine Segment's Arena**
 - Mission Application
 - Infrastructure or Common Service
- **Register Segment with the DII AF Office.**
- **Use current Segmentation Tool Kits for to Segment NT applications for Version 3.1, 3.2, 3.3.**
 - Full Segmentation process for GOTS
 - Abbreviated Segmentation for COTS

How to Segment Process



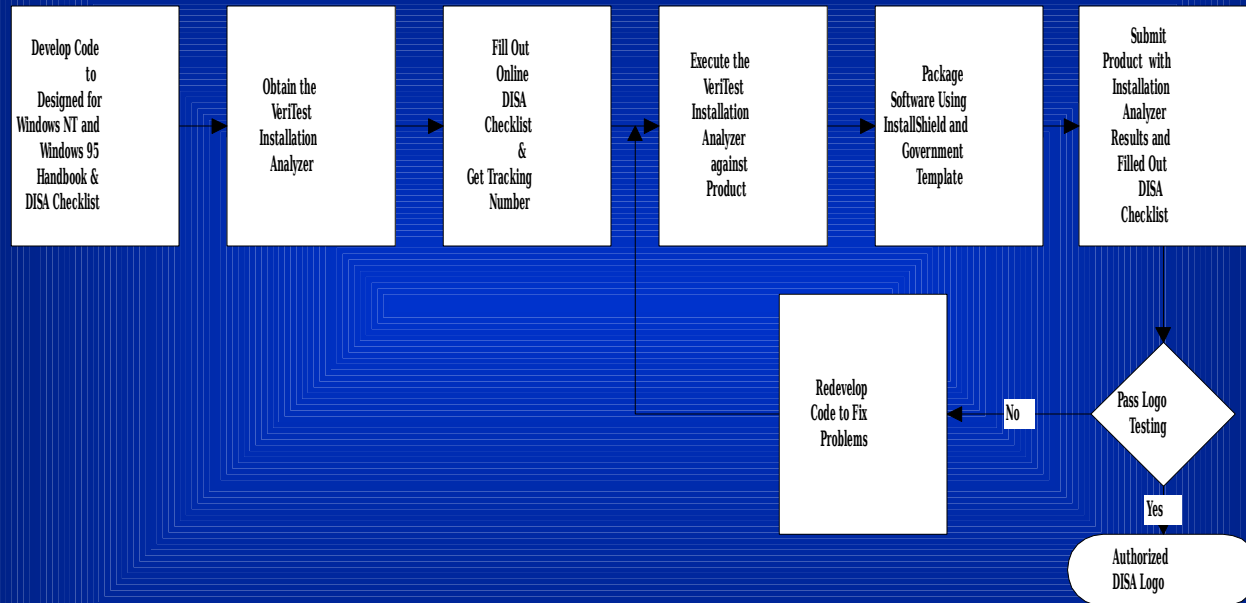
Future Segmentation Process

- **InstallShield for DII COE NT v4.0**
 - DoD Templates to Preserve DII COE Configuration
 - Vendor has Expressed Interested in Building DoD Templates

Compliance Assurance

- **Current Guidance**
 - Self Assessment (e.g., Checklist Assessment)
 - Developers should be Encouraged to Use VeriTest Logo Compliance Freeware (Analyze.exe)
 - Integration Testing (e.g., CFI and DCTF Testing)
- **Update Documentation (I&RTS Checklist, etc.)**
 - Checklist Must Reflect Logo and DII COE Requirements
- **Option**
 - Logo Certification
 - DISA (CFI, DCTF) Certification
 - Out-Source Certification (e.g., Veritest)

DISA Workstation Development Logo Process Alternative



NTAG Previous Option Recommendation

- **Define a Comprehensive Certification Process for DII COE NT Segments that Mandates Comprehensive Testing of the Requirements Identified in Recommendation 1.**
- **Allow Developers the Option of Using of Either a Revised Government or Microsoft NT Logo Program Certification Process.**

NTAG Previous Option Recommendation

- **DII COE Chief Engineer select and implement the Government Certification Process. Options in the White Paper Include:**
 - Outsourcing Certification Testing to VeriTest and Microsoft.
 - Outsourcing Certification Testing to Third Parties using the Current Logo Certification Software.
 - In-House Certification using the Current Logo Certification Software.
 - In-House Certification using the Government developed. Certification Software that would be Similar to the Current Logo Certification Software.

Segmentation

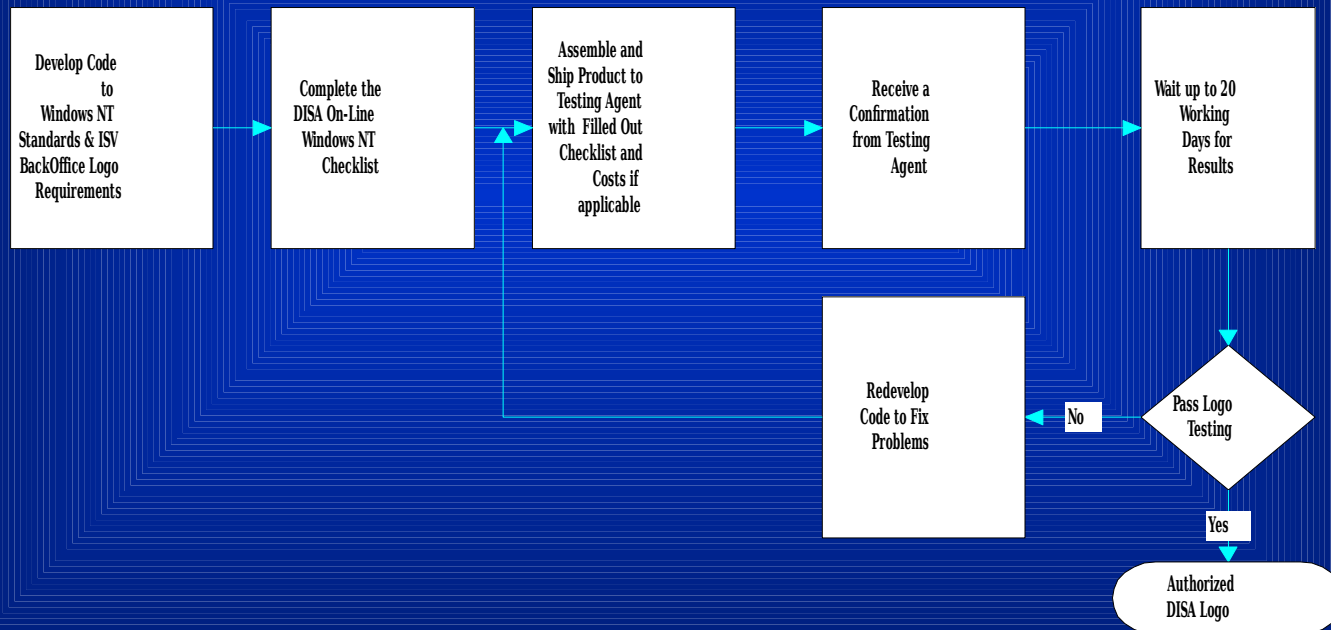
15 & 16 January 1998

Questions?

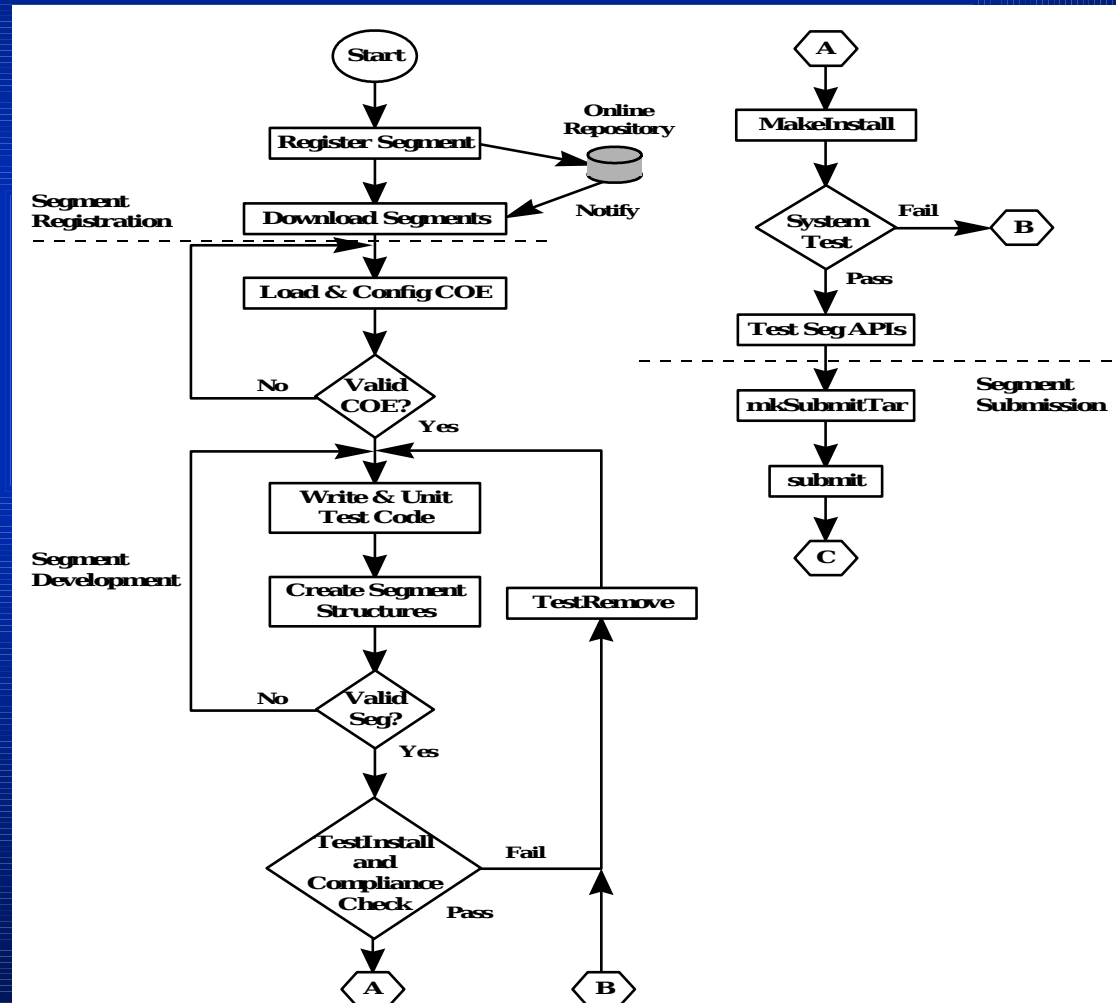
Mr. Robert P. Blaisdell
Electronic Systems Center DII AF
email: rpb@mitre.org
(781) 271-5757

Backup Slides

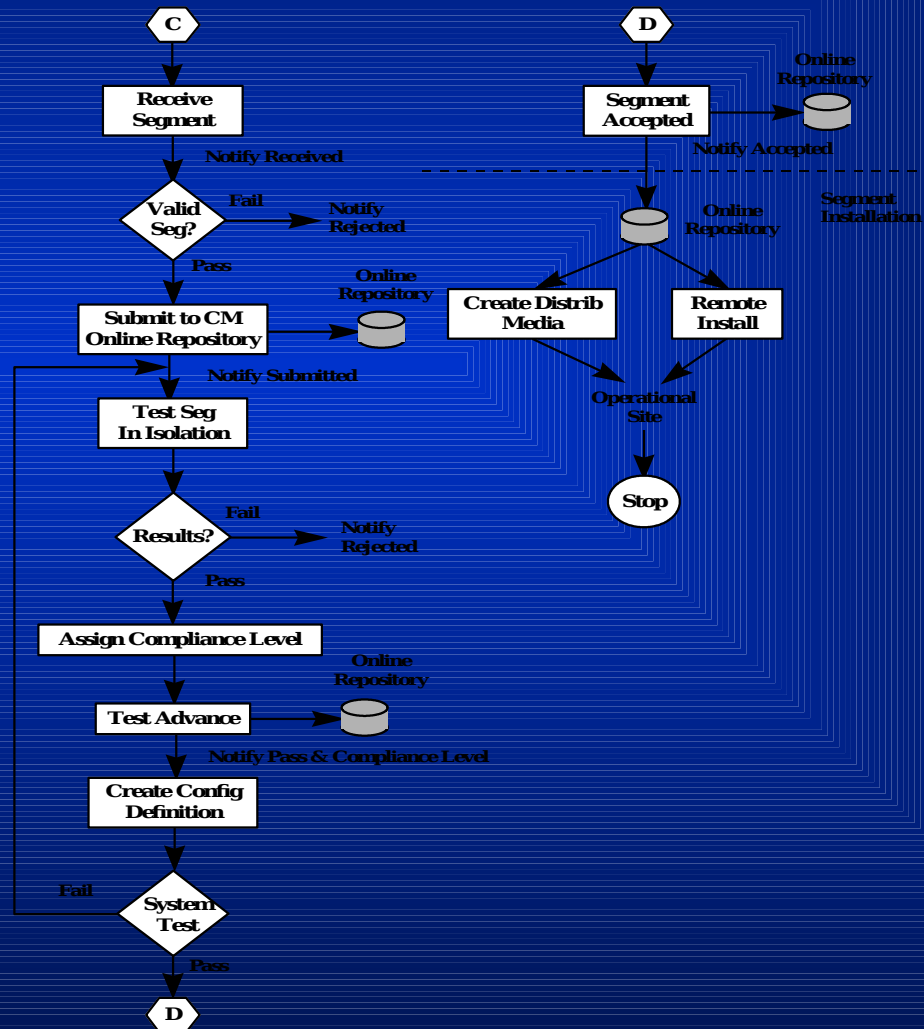
Revised Client/Server Logo Process Alternative



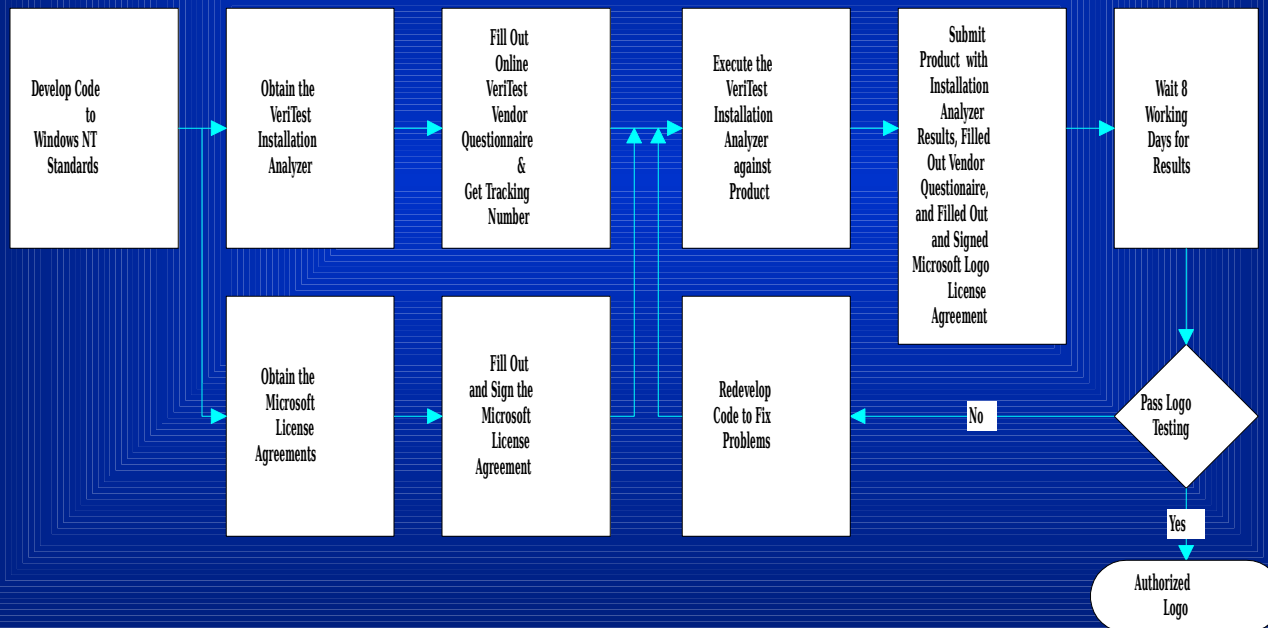
DISA Segmentation Process (Part 1)



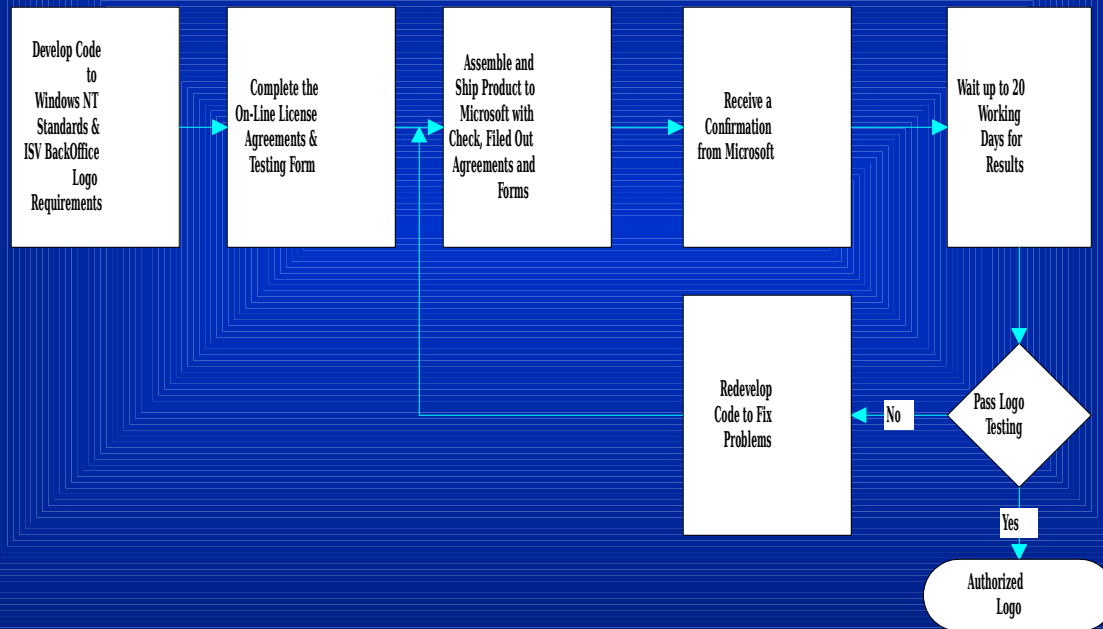
DISA Segmentation Process (Part 2)



Designed for Windows NT and Windows 95 Logo Program Process



Independent Software Vendor Designed for BackOffice Logo Program Process



Installation Requirements

User Experience

- The product must have an installation program with a graphical Setup program that executes in a 32-bit Microsoft Windows operating system.
- The installer must either lead the user through the process in a user attended mode, (not just instruct the user to copy or decompress files) or provide a facility for a 'silent' or unattended install requiring no user input. Preferably, (though not required) the application will provide both options.
- The installer must automatically detect the version of Microsoft Windows NT and/or Windows 95 running and install the correct version of your product automatically. This must be a seamless experience for the user.

Installation Requirements

User Experience

- All shortcuts created by the installation must be viable, that is, not result in "file not found" or other error conditions when activated.
- Products distributed on CD-ROM must utilize the AutoPlay feature to begin setup or launch the program itself the first time the application is run. It is up to the vendor whether AutoPlay is enabled on subsequent insertions of the CD-ROM. In the case of products distributed on multiple CD-ROMs, subsequent CD-ROMs must either utilize the AutoPlay feature or behave during install like a subsequent floppy disk (continue install without prompting user for action.) It is not acceptable to require the end user to Start, Run the CD-ROM during installation.

Installation Requirements

User Experience

- The installer for media other than CD-ROM must provide the ability to launch the installer through Add/Remove Programs in the Control Panel.
- The application must default installation to a directory (or directories) under "drive:\Program Files". The installer must query the registry for this directory path to ensure appropriate install if, for example, a user has renamed this directory on their machine, or if the application is being installed onto a non-English language, localized version of Windows.
- The application's installer must not install application DLLs or executables under the system root.

Installation Requirements

User Experience

- **The installer must check before starting any operation to ensure that it can complete that operation. The following are the requirements based on this principle.**
 - **Installers must fail gracefully in the case where a system component cannot be replaced because file system security prevents an existing file from being overwritten. Installers that need to overwrite system files, for example, should check up-front if the current user is a member of the administrators local group. This will avoid confusion later by preventing the user from getting file copy errors from which there is no recovery.**
 - **Applications must be tested on Windows NT under non-administrator accounts. An application must run under a user account but that user must not be able to change the application setup configuration.**

Installation Requirements

User Experience

- When installing the application, the installer must check for the user privilege level. If the user is not an administrator and the application will work but with limited functionality, the installer must warn the user that only limited functionality will be available since they do not have admin privileges, and the installer must allow them to discontinue the installation.

Installation Recommendations User Experience

- **Applications should not assume that directory names will be in the English language, or unchanged by the user (e.g., "Program Files".) Applications should query the registry directly to obtain the proper, language specific directory names. Vendors should be aware that English versions of their application might be installed onto a non-English language version of Windows. In this instance a search for the English language directory string "Program Files" would not be successful & would result in a failed install. For other folders the proper paths can be queried via the SHGetSpecialFolderLocation() function with CSIDL flags:**

Installation Recommendations User Experience

- **User settings should be maintained across application version changes. It's also recommended that user settings be able to be moved from machine to machine. Examples of user settings include customized toolbars, any dynamic, user-generated lists or other user specific options. For information about supporting a Networked or Managed environment refer to section 6.2.4.**
- **If the application can be sure it is installing on a single user machine, it should collect user name or Personal ID (PID) information the first time the application is run, not during install.**

Installation

Recommendations

CSIDL Flags

- **#define CSIDL_PROGRAMS 0x0002**
- **#define CSIDL_PERSONAL 0x0005**
- **#define CSIDL_FAVORITES 0x0006**
- **#define CSIDL_STARTUP 0x0007**
- **#define CSIDL_RECENT 0x0008**
- **#define CSIDL_SENDTO 0x0009**
- **#define CSIDL_STARTMENU 0x000b**
- **#define CSIDL_DESKTOPDIRECTORY 0x0010**
- **#define CSIDL_NETHOOD 0x0013**
- **#define CSIDL_TEMPLATES 0x0015**
- **#define CSIDL_COMMON_STARTMENU 0x0016**
- **#define CSIDL_COMMON_PROGRAMS 0x0017**
- **#define CSIDL_COMMON_STARTUP 0x0018**
- **#define CSIDL_COMMON_DESKTOPDIRECTORY 0x0019**
- **#define CSIDL_APPDATA 0x001a**
- **#define CSIDL_PRINTHOOD 0x001b**

Installation Requirements

Using the Registry

- **Native data file types (if applicable) must be registered as follows:**

[HKEY_CLASSES_ROOT]

.(file type extension)

(Default) = REG_SZ:FileTypeID

See the "Windows Interface Guidelines for Software Design," chapter 10 (available from MS Press), for further details on registering file types and application data.

- **The application installer must register *all* shared components designed to be uninstalled under the following Registry key:**

**[HKEY_LOCAL_MACHINE]\SOFTWARE\Microsoft\Windows
\Current Version\SharedDLLs**

Installation Requirements

Using the Registry

- The installer must *not* register components that are self-registering, such as DirectX and Direct3D. The installers for these components will register their components as needed.
- Core components must *not* be refcounted in the registry by application installers. Please see <http://www.microsoft.com/windows/thirdparty/winlogo/> for the most up-to-date and complete list of Windows 95 and Windows NT core components.
- The application must not add information to Win.ini or System.ini.

Installation Recommendations

Using the Registry

- **Applications should provide the following set of information in the registry. This information will be used by the Add/Remove Programs Control Panel to provide a central place for information about the application.**
 - The registry values in the table below should be written under the following key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\NameofApplication

Registry Table Entries

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\Name of Application

Name of Value	Type	Contains
• DisplayName	REG_SZ	Display Name of Application.
• UninstallPath	REG_SZ*	Full path to the Application's uninstall program.
• ModifyPath	REG_SZ*	Full path to the Application's setup/modify program.
• InstallLocation	REG_SZ*	Full path where the Application is located (folder or *.exe).
• InstallSource	REG_SZ*	Location where the Application was installed from.
• DisplayVersion	REG_SZ	Displayed version number of Application.
• VersionMajor	DWORD	Major Version Number of Application.
• VersionMinor	DWORD	Minor Version of Application.
• Publisher	REG_SZ	Publisher/Developer of Application.
• ProductID	REG_SZ	Product ID of installed Application.
• RegOwner	REG_SZ	Registered owner of Application.
• RegCompany	REG_SZ	Registered company of Application.
• HelpTelephone	REG_SZ	Telephone number for support/help.
• HelpLink	REG_SZ*	Full path of Help File or Help URL for Application.
• URLUpdateInfo	REG_SZ*	URL of Update Information for Application.
• URLInfoAbout application's	REG_SZ*	URL that provides a link to the publisher's home page or the home page.

NT

Advisory

Installation Requirements

System & Shared Components

- Applications must not overwrite core components with older versions of the components. The vendor must check <http://www.microsoft.com/windows/thirdparty/winlogo.htm> for the most up-to-date and complete list of Windows 95 and Windows NT core components.
- The vendor must run Pretest the application using VeriTest's Install Analyzer & submit the report results with the Vendor Questionnaire. Improper handling of core and shared components is the number one reason applications fail Logo testing on the first try and must retest. The Install Analyzer provides automated testing of compliance in this area. The Install Analyzer can be found at http://www.veritest.com/installation_analyzer.htm

Installation Recommendations

System & Shared Components

- **It is strongly recommended that applications do not write anything to the \system32 directory. On Windows NT 5.0 the OS will support a mode in which this directory will be read-only and unavailable to applications & installers. This will be part of the Zero Administration Windows effort to reduce DLL conflicts and other shared component problems.**
- **System Reboots should be avoided (this is strongly recommended.) If the system must be rebooted (for example, after installing a service pack for the OS) setup should gracefully resume where it left off.**

Installation Recommendations

System & Shared Components

- It is strongly recommended that if your application requires updated system DLLs or other core components you must ship a self-extracting EXE from Microsoft which will handle installation of these components. Applications are strongly discouraged from installing system components or install into Windows\system32 by any other means than Microsoft provided service packs. *(MS will distribute all new DLLs by means of service packs only. Vendors should distribute self-extracting EXEs from Microsoft rather than the latest DLLs.)*

Installation Recommendations Support of a Network or Managed Environment

- **The following behaviors are recommended for Installation in a Networked or Managed environment.**

For more information about how your applications can support a networked or managed environment refer to <http://www.microsoft.com/windows/zaw>

- **Registry Use: It is recommended that HKCU settings that are application defaults or part of an administered profile should not be written to the registry until the user modifies them (through setting/option changes, etc.). Until such a time, the settings should reside within the application. Therefor any HKCU keys that are written to the registry should be different than those that reside in the application (defaults).**

Installation Recommendations

Support of a Network or Managed Environment

- **Setup should support network installations that enable network administrators to add additional components to network installs. (for example, third-party tools, add-ons, etc)**
- **Setup should support creating a disk image that can be installed onto multiple machines. In this situation machine & user specific settings should be collected the first time the application is run.**
- **Setup should not install shared files in non-shared volumes. Files shared by multiple users or applications should reside in a location that is common and accessible.**
- **Applications should support silent installs & 'push' strategies in managed environments.**

Installation Recommendations

Support of a Network or Managed Environment

- Applications should be capable of running directly from source in the case of CD or network installs. This should be done with no local machine footprint. (Local use of the registry is considered 'zero-footprint' where unavoidable.)

Installation

Recommendations

Use of Policy in a Managed Environment

- It is recommended that applications include a system policy *template* to enable administrators to remotely manage the roll-out and use of the application. By utilizing a system policy template, global changes to registry parameters can be applied to all users, or a subset of users at the next interactive logon. It is recommended that the application offer the administrator a template of registry settings to choose from in customizing that application's rollout or behavior on user's machines. This template is referred to as an ADM file. The administrator (or software developer) can mandate a value, remove a value, or disregard the current state of a particular value at each interactive user logon. For example, a vendor can provide a template to administrators that can set the default server locations in each user's personal preferences.

Installation

Recommendations

Use of Policy in a Managed Environment

- **Recommended considerations for developing applications to support use of system policy:**
 - **It is recommended that vendors include an ADM file with their application or make such a file available for download from the vendor's web site.**
 - **If the application utilizes the Run or Find dialog, support should be added for the scenario when a system policy is in effect that disables the user's capability to use these features.**
 - **If the application is a shell extension, it should support the system policy NoViewContextMenu, which disables the context menu on the desktop or within the Explorer interface.**

Installation

Recommendations

Use of Policy in a Managed Environment

- If the application stores paths in the registry, the application should utilize **ExpandEnvironmentStrings** to support variables as part of the path, such as **%USERPROFILE%**. Hard-coded path names should not be used.
- When spawning an application, use the Win32 API **ShellExecute** instead of **CreateProcess**. As a component of system policies, administrators can define which applications cannot be started from the Explorer interface. Utilizing **ShellExecute** allows this check to occur before starting the application.

Installation

Recommendations

Use of Policy in a Managed Environment

- **Policy keys added to the registry should be created under either HKLM\Policy or HKCU\Policy and should follow the "software" key style, for example:
HKLM\Policy\companyname\NameofApplication\<values>. If these keys do not already exist, they should be created following this format.**

Installation Recommendations

Support Migration from Windows 95

An application should fully operate after the operating system is upgraded (for example, from Windows 95 to Windows NT). To do so, applications should comply with the following:

- **When installing on one 32-bit Microsoft Windows operating system, the installer should provide an option to the user whether to install all additional binaries that would be required to make the program operate on the other operating system. The application should determine at run time which components to use. For example, if the application requires a different DLL on Windows NT than on Windows 95, the application should install both and pick the one to actually load and use at application startup time. If the application requires a different .exe for each operating system, then write a stub .exe that calls the correct .exe based on a run-time version check (GetVersionEx() Win32 API).**

Installation Recommendations Use of Policy in a Managed Environment

- **Review your product to ensure that it:**
 - **Does not write to different registry locations on Windows 95 and Windows NT (registry values may also be different)**
 - **Does not install different files (DLLs, EXEs, etc.) on the two operating systems (or additional files on Windows 95)**
 - **Does not install different versions of files that are common to both Windows 95 and Windows NT**
 - **Does not make operating system specific calls on Windows 95 that are not available on Windows NT**

Installation Recommendations

Windows 95 to NT 5.0 Migration

- **Future versions of the Designed for Windows Logo will require that the Windows 95 installation of the product is functional after a user upgrades their Windows 95 machine to Windows NT 5.0 or later. To accomplish this, vendors should ensure that the differences between the Windows 95 and the Windows NT versions of their applications are minimal.**
- **If an application needs to move or modify registry entries, move files or system settings in order to get the Windows 95 version of that application to run on Windows NT, vendors should write a migration DLL as described in the Windows 95 to Windows NT 5.0 Migration Kit. The migration DLL should perform all tasks necessary to make the Windows 95 version of the application functional on Windows NT 5.0. This should include modifying or adding new registry entries and copying new files to their proper locations on Windows NT 5.0.**

Installation Recommendations

Windows 95 to NT 5.0 Migration

Support

- **During an upgrade it is strongly recommended that the migration DLL not require user interaction or display a User Interface to the end-user.**
- **For new applications, it is recommended that vendors create a migration DLL and place it on their product media so that users can find them in case they need to upgrade their computers from Windows 95 to Windows NT 5.0 or later.**

Uninstall Requirements

User Experience

- The product must provide a fully automated uninstaller that removes the program files, folders, and Registry entries for the 32-bit Microsoft Windows 95 and Windows NT operating systems.
- The uninstaller must be accessible through Add/Remove Programs on the Control Panel and must operate properly from the Control Panel.
- The uninstaller must be properly registered and must appear under Add/Remove Programs in the Control Panel. The method for this registration is:

```
[HKEY_LOCAL_MACHINE]\SOFTWARE\Microsoft\Windows\Current  
Version\Uninstall\YourProductName
```

```
DisplayName=REG_SZ:<your product name and version number>
```

```
UninstallString=REG_SZ: c:\apps\myapp\uninstll.log /h
```

Uninstall Requirements

User Experience

- **All files and folders copied onto the hard disk must be removed, with certain exceptions.**
 - **Resources that other programs might use, such as sharable DLLs, sharable fonts, and sharable Registry entries.**
 - **It is better to err on the side of safety regarding other applications. If you are not sure whether removing a DLL might harm other applications, it is better to leave it behind. However, you must explain everything you leave behind when you submit your application for testing. The proper place to do this is in the Vendor Questionnaire.**
- **The uninstaller must remove all advertisements and shortcuts placed anywhere in the Start Menus by its installer that are associated with the component that is being removed.**

Uninstall Recommendations

User Experience

- **On Windows NT a user must not be able to uninstall an application if that user does not have adequate security permissions to complete the uninstall. In this situation the application must provide an appropriate error message such as "You do not have adequate security permissions to successfully uninstall this application. Please contact your system administrator." A system administrator should be able to uninstall all applications that were installed (by a user or administrator) using the default installation.**

Uninstall Requirements

Registry, System & Shared Components

- The uninstaller must remove all Registry entries (with the exception of keys that might be shared by other programs).
- The uninstaller must *not* decrement or remove any core component, in particular Microsoft Foundation Class Library (MFC) DLLs, as well as ODBC & DAO DLLs. Please see <http://www.microsoft.com/windows/thirdparty/winlogo/> for the most up-to-date and complete listing of all Windows NT and Windows 95 core components. The core component list is updated regularly. It is the vendor's responsibility to check the web for the most up to date list.

Uninstall Requirements

Registry, System & Shared Components

- **The uninstaller must accurately decrement the count on all components your application uses that are installed as shared components.**
- **If the Reference Count is at zero (0) and you will not be removing the component, you must leave the Reference Count at 0. This requirement prevents installers from inaccurately incrementing the Shared Count if the component was on the system, but was not registered.**
- **The uninstaller must remove itself.**

Uninstall Recommendations Registry, System & Shared Components

- **If your installer finds a shared component already on the system and not registered, the SharedDLL (see section 6.2.2) count should be incremented by 1 plus the number of clients being installed. For example, if you install your application with 3 clients using a shared component, your installer will bump the SharedDLL count by 3. But if the shared component was already on the system and no SharedDLL exists for it (that is, the previous installer did not create the refcount), then set the SharedDLL count to 4. That way, when your application is uninstalled, it leaves the shared component on the system with a refcount of 1.**

Uninstall Recommendations

Support of Managed Environment

- **Administrators should be able to uninstall and unadvertised (remove shortcuts) the application by means of a centrally managed Group Policy Template or through changes to user Profiles. Refer to section 6.2.5 for more information on the use of Policy & Policy Templates.**

Uninstall Recommendations

Managed Installation Services

- **The application should make appropriate use of user Profiles. In the NT 5.0 timeframe this means:**
 - **Maintain strict separation of user data from application bits whenever possible.**
 - **On Windows NT, applications should use the document storage folder in the user's Profile to store user data. The Profile path can be queried via the SHGetSpecialFolderLocation() function with the CSIDL_PERSONAL flag.**
 - **On Windows NT applications should use the ...\\winnt\\Profiles\\username\\Application Data and ...\\winnt\\Profiles\\username\\Personal folders.**

Uninstall Recommendations

Managed Installation Services

- **User data should by default be stored in the user Profile**
- **If the application settings are no longer part of a Profile, the application should recover gracefully and regenerate the needed settings based on the application's defaults.**